01_virtual

This code features a base class with a pure virtual function and a derived class which implements this function. Note the member functions of the base class; `get_x` and `set_x`.

```cpp
/**
 * @file 01_virtual.cpp
 * @author 0×ca7 (0×ca7.github.io)
 * @brief a simple C++ class with pure virtual function
 *                      and inheritence
 * @version 0.1
 * @date 2022-12-27
 *
 * @copyright Copyright (c) 2022
 */

#include <iostream>
#include <stdio.h>

class Base {

        int x;
public:
        // this is a pure virtual function
        virtual int calculation() = 0;
        int get_x();
        void set_x(int x);
};

int Base::get_x() {
        return this->x;
}

void Base::set_x(int x) {
        this->x = x;
}

class Derived: public Base {
        int xx;
public:
        // implementation of calculation
        int calculation() {
                int x = 0;
                x = (this->xx + 0×deadbeef) % 0×baadf00d;
                return x;
```

```
        }
};

int
main(void)
{
        int i = 0;
        Derived d;

        d.set_x(1337);
        i = d.get_x();

        printf("derived x: %d\n", i);

        i = d.calculation();
        printf("derived x after calculation: %d\n", i);

        return 0;
}
```

First interesting part is the constructor of Derived. The assembly is shown below.

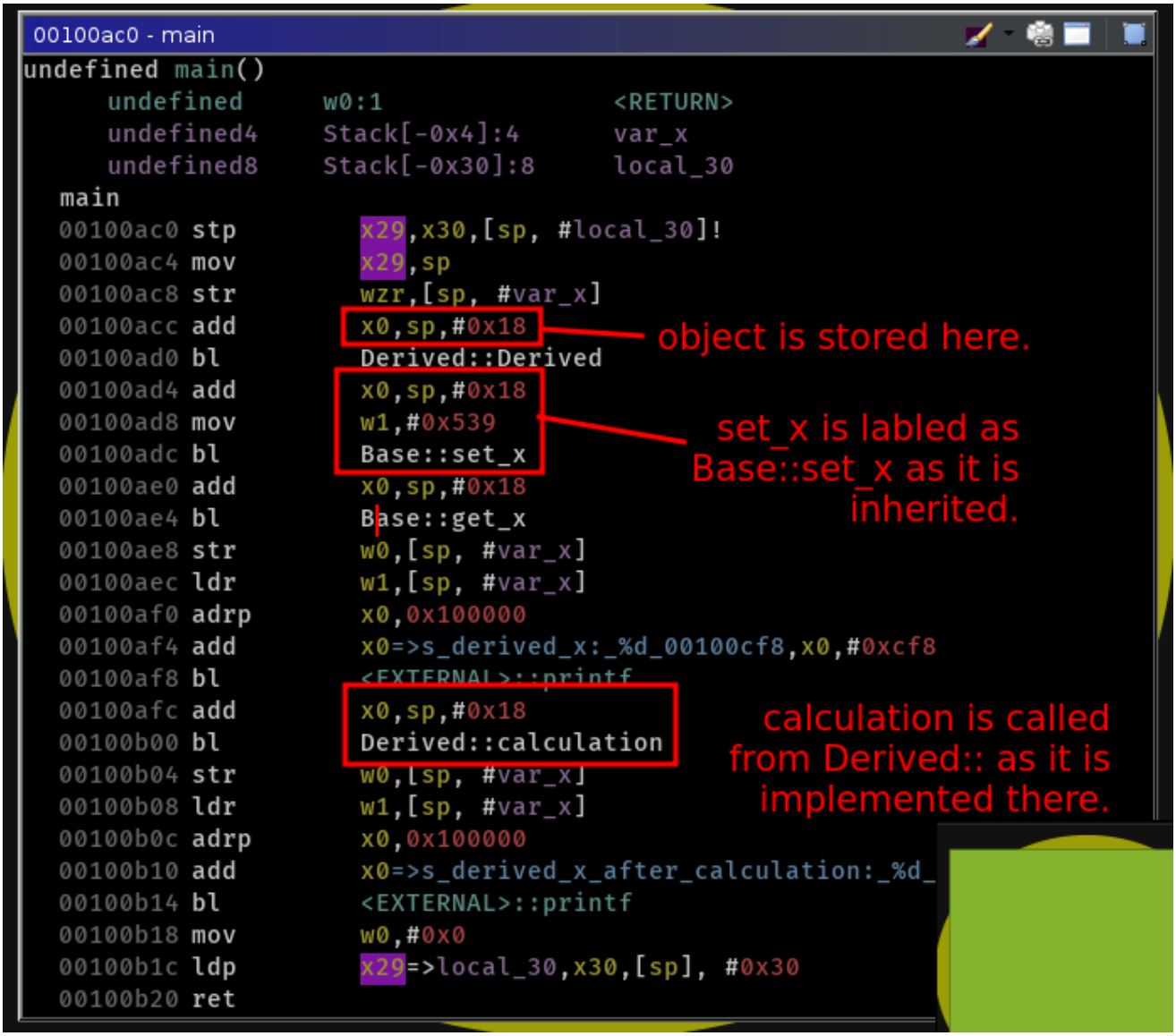
```
undefined __thiscall Derived(Derived * this)
      undefined     w0:1              <RETURN>
      Derived *     x0:8 (auto)       this
      undefined8    Stack[-0×8]:8     var_this
      undefined8    Stack[-0×20]:8    local_20
  _ZN7DerivedC1Ev
  _ZN7DerivedC2Ev
  Derived::Deriv
  00100c18 stp        x29,x30,[sp, #local_20]!
  00100c1c mov        x29,sp
                      ; store the this pointer.
  00100c20 str        this,[sp, #var_this]
  00100c24 ldr        this,[sp, #var_this]

                      ; call the constructor of the base class.
  00100c28 bl         Base::Base
                      ; calculate the address of the implementation
                      ; of the virtual function.
  00100c2c adrp       this,0×111000
  00100c30 add        x1,this,#0×d68
  00100c34 ldr        this,[sp, #var_this]
                      ; store a pointer to the implementation of the
```

```
virtual
                    ; function so it can be used in the class.
  00100c38 str        x1⇒PTR_calculation_00111d68,[this]
  00100c3c nop
  00100c40 ldp        x29⇒local_20,x30,[sp], #0×20
  00100c44 ret
```

Now, the constructed *Derived* object has a pointer to it's implementation of the method `calculation`. Here is the base class constructor:

```
undefined __thiscall Base(Base * this)
      undefined     w0:1              <RETURN>
      Base *        x0:8 (auto)       this
      undefined8    Stack[-0×8]:8     var_this
  _ZN4BaseC1Ev
  _ZN4BaseC2Ev
  Base::Base
  00100bf4 sub        sp,sp,#0×10
                  store the this pointer.
  00100bf8 str        this,[sp, #var_this]
  00100bfc adrp       this,0×111000
                  get the offset of "cxa_pure_virtual"
                  at 0×111000 + 0×d80 = 0×111d80 to x1
  00100c00 add        x1,this,#0×d80
  00100c04 ldr        this,[sp, #var_this]
                  store this address.
  00100c08 str        x1⇒PTR___cxa_pure_virtual_00111d80,[this]
  00100c0c nop
  00100c10 add        sp,sp,#0×10
  00100c14 ret
```

The main function looks like this; note the annotations.


00100ac0 - main
undefined main()
undefined w0:1 <RETURN>
undefined4 Stack[-0x4]:4 var_x
undefined8 Stack[-0x30]:8 local_30
main
00100ac0 stp x29,x30,[sp, #local_30]!
00100ac4 mov x29,sp
00100ac8 str wzr,[sp, #var_x]
00100acc add x0,sp,#0x18
00100ad0 bl Derived::Derived
00100ad4 add x0,sp,#0x18
00100ad8 mov w1,#0x539
00100adc bl Base::set_x
00100ae0 add x0,sp,#0x18
00100ae4 bl Base::get_x
00100ae8 str w0,[sp, #var_x]
00100aec ldr w1,[sp, #var_x]
00100af0 adrp x0,0x100000
00100af4 add x0=>s_derived_x:_%d_00100cf8,x0,#0xcf8
00100af8 bl <EXTERNAL>::printf
00100afc add x0,sp,#0x18
00100b00 bl Derived::calculation
00100b04 str w0,[sp, #var_x]
00100b08 ldr w1,[sp, #var_x]
00100b0c adrp x0,0x100000
00100b10 add x0=>s_derived_x_after_calculation:_%d_
00100b14 bl <EXTERNAL>::printf
00100b18 mov w0,#0x0
00100b1c ldp x29=>local_30,x30,[sp], #0x30
00100b20 ret
object is stored here.
set_x is labled as Base::set_x as it is inherited.
calculation is called from Derived:: as it is implemented there.